Empowering the Weakest Link in Today's Technology with: Cyber-security Protection by Means of Password Security

Author: Lazaro R. Serrano Malave
Advisor: Dr. Nelliud Torres
Department of Computer Science

Abstract

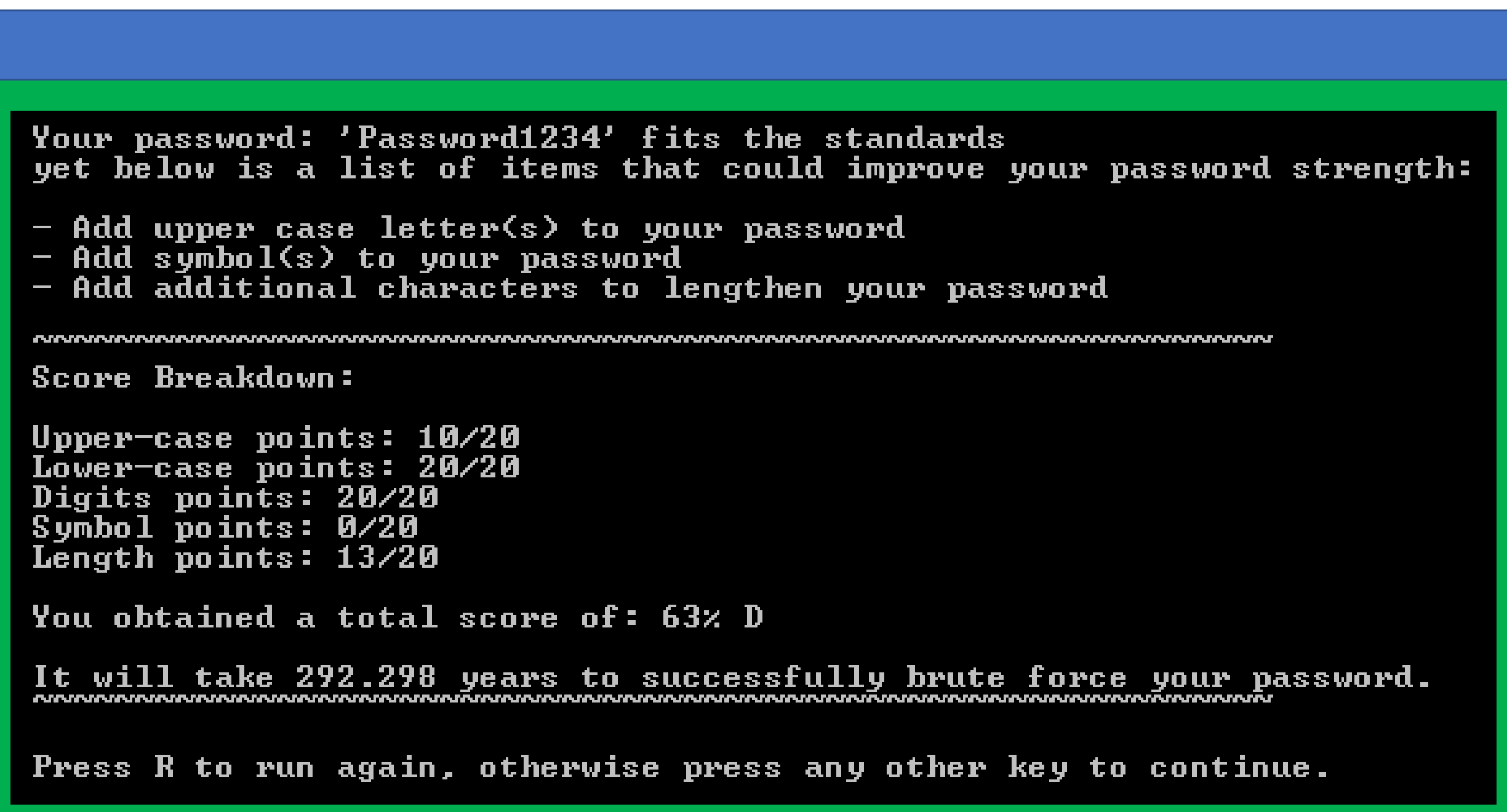
No matter how expensive or how advanced the technology inside your organization is, there is always going to be the human factor. In this project I develop a program written in C++ that will help not only IT professionals but also the general public that use passwords to log in to their work environment. This software will prompt the user to enter a password and the program will then give the user a password score while suggesting the user to add certain ASCII characters to help strengthen their password and try again, as well as displaying how many years it would take for a super computer to crack the user's password. With 81% of hacking-related breaches being generated because of stolen and/or weak passwords, it is important for users to have a well-constructed password to avoid being the target of a criminal hacker.

Introduction

We are currently living in an era where technology is found in every aspect of our lives. And with great advances comes great responsibility. Wherever there are people working hard to achieve their goals there will always be one other group of people with a different goal in mind, ransack organizations and individuals alike. These individuals will try to penetrate your systems with the purpose of stealing your information for their own personal use or to trade stolen information for resources. These individuals are known as malicious hackers. To defend off against these types of groups, individuals and organizations alike will need to not only be highly trained against emerging threats, yet they will need to acquire certain tools and software that will help protect their computer systems from the theft of information and data.

Guidelines

When running this software, it is important to know your main goal. The main goal for the user executing this code is to create a password string while modifying that existing password by applying the suggestions the software displays. When typing in the string, the analyzer will prompt the user to enter a string with a minimum of eight characters, after the eight characters or more string has been input, the user will then hit enter to pass onto the next screen, which will display suggestions on how that user can modify their existing password to create an even stronger password.



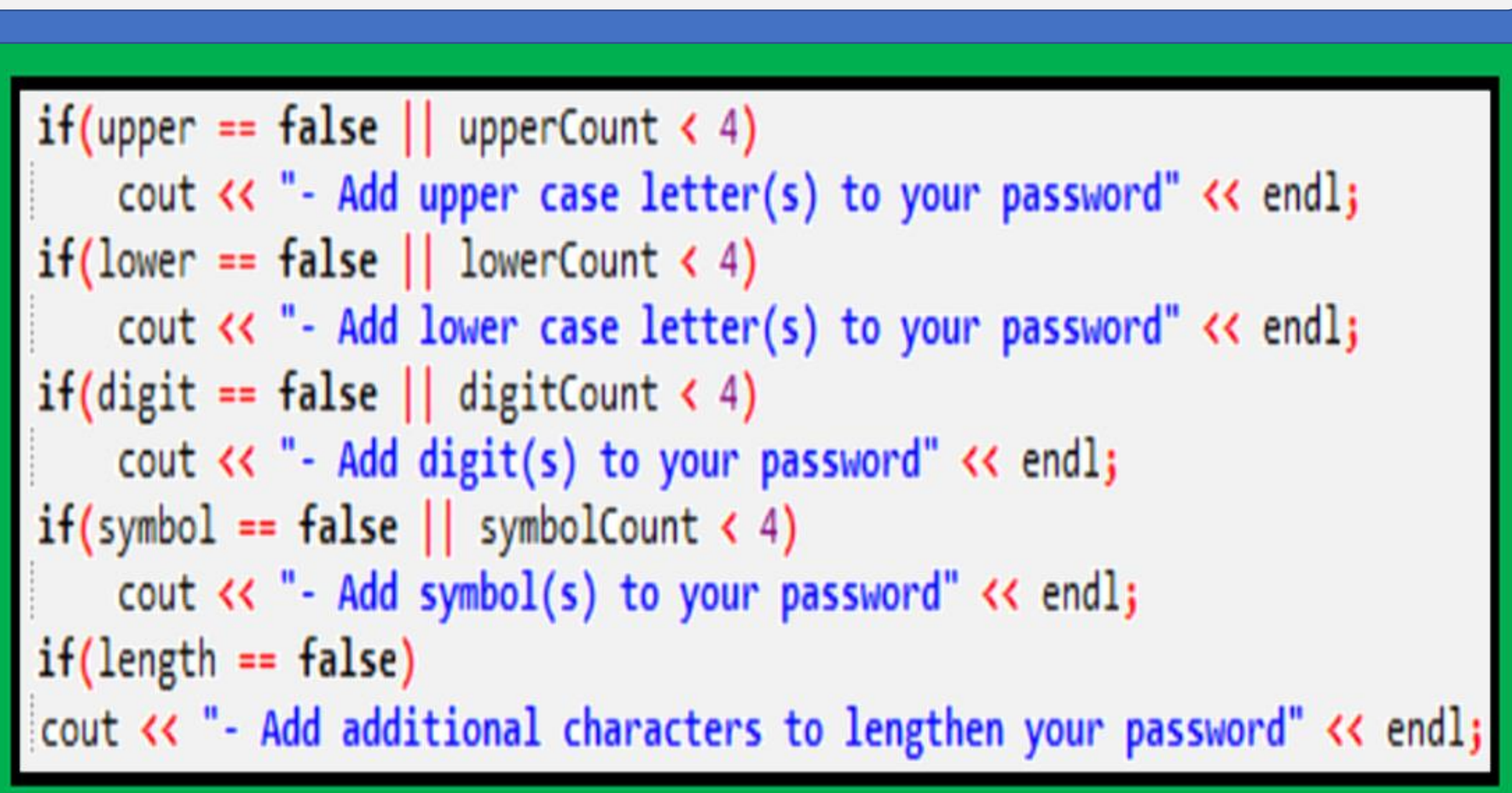
```
Your password: 'Password1234' fits the standards
yet below is a list of items that could improve your password strength:

- Add upper case letter(s) to your password
- Add symbol(s) to your password
- Add additional characters to lengthen your password

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Score Breakdown:

Upper-case points: 10/20
Lower-case points: 20/20
Digits points: 20/20
Symbol points: 0/20
Length points: 13/20

You obtained a total score of: 63% D

It will take 292.298 years to successfully brute force your password.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Press R to run again, otherwise press any other key to continue.
```

Suggestions made to the User after Running the Password Analyzer

This screen will then display a password score from a minimum of 0% being the worst score up until a 100% which will be the highest score possible. In addition to the password score, the Password Analyzer will output to the user how many years it would take for that password to be cracked using the worlds most sophisticated super computers.

Methodology

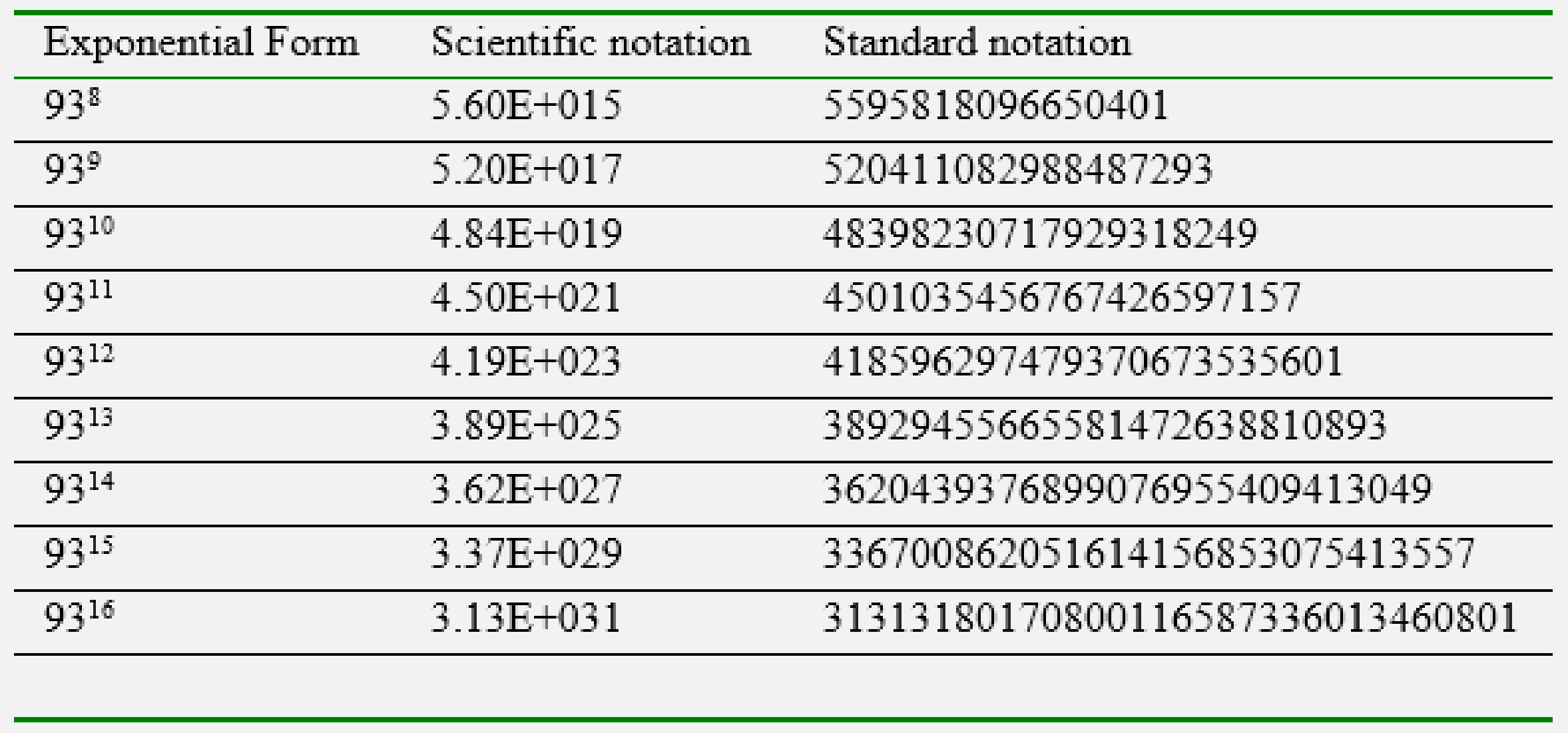
In this project I discuss how a password analyzer can help all organizations, from small companies to large corporations. The way this password analyzer works is by first asking a user to enter a password that has at least an eight-character length, if the length falls short then it will prompt the user to re-enter his or her password once again. Once the user enters a valid password string the program will then provide: suggestions on how to edit your current password to make it an even more secure one.



```
if(upper == false || upperCount < 4)
   cout << "- Add upper case letter(s) to your password" << endl;
if(lower == false || lowerCount < 4)
   cout << "- Add lower case letter(s) to your password" << endl;
if(digit == false || digitCount < 4)
   cout << "- Add digit(s) to your password" << endl;
if(symbol == false || symbolCount < 4)
   cout << "- Add symbol(s) to your password" << endl;
if(length == false)
cout << "- Add additional characters to lengthen your password" << endl;
```

Code Written in C++ on How the Software Alerts the User to Modify Entered Password String

There are a total of 128 unique ASCII characters, yet only 95 are printable characters. The amount of printable characters implemented in my software will have a total of 93 unique printable ASCII characters. These 93 characters are composed of the following: lower case letter characters which contain a total of 26 unique characters, upper case letter with a total of 26 unique characters, digit characters with a total of 10 unique characters and special symbols which contain a total of 31 unique characters [1].



Exponential Form	Scientific notation	Standard notation
93^{8}	5.60E+015	5595818096650401
93^{9}	5.20E+017	520411082988487293
93^{10}	4.84E+019	48398230717929318249
93^{11}	4.50E+021	4501035456767426597157
93^{12}	4.19E+023	418596297479370673535601
93^{13}	3.89E+025	38929455665581472638810893
93^{14}	3.62E+027	3620439376899076955409413049
93^{15}	3.37E+029	336700862051614156853075413557
93^{16}	3.13E+031	31313180170800116587336013460801

Distinct Possible Combinations for a Fixed String Length with 93 Unique Printable Characters in the ASCII Set

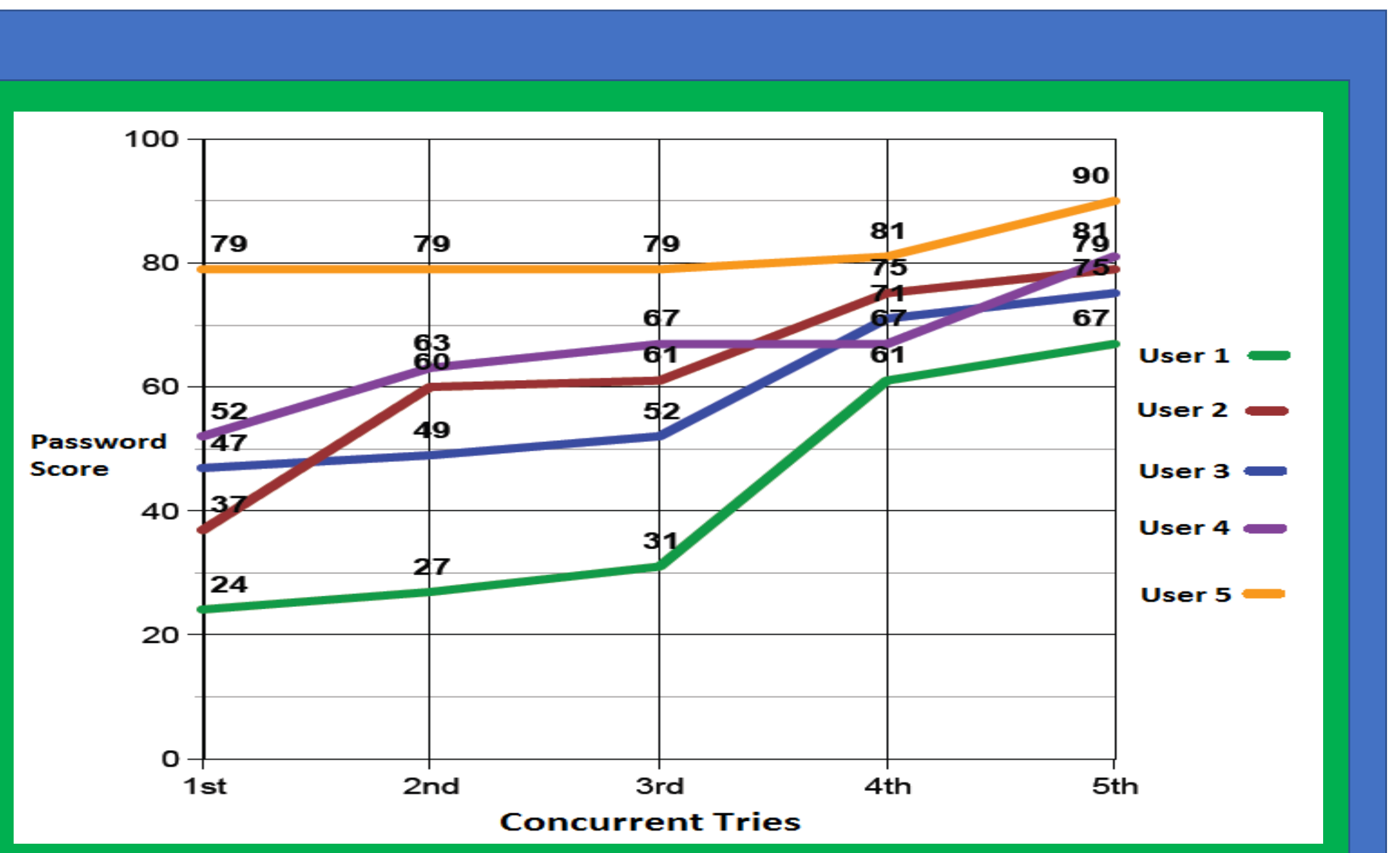
The first part of the formula to calculate how many years it would take to successfully brute-force a password contains the following: (93^(Password Length)) which will be the amount of unique characters which is 93 elevated to the power of the password length. The number three hundred fifty billion (350,000,000,000) which is the amount of guesses the world's fastest super computer can cycle each second [2]. The last segment of the formula (60*60*24*365) will consist in the conversion from seconds to years where there are 60 seconds in a minute, 60 minutes in an hour, 24 hours in a day and 365 days in one year, that is a total of 31,536,000 seconds each year.

$$\frac{93^{\text{Password Length}}}{(350{,}000{,}000{,}000) \times 60 \times 60 \times 24 \times 365}$$

Formula to Calculate in Years, how much Time it Would Take to Successfully Brute-force a User's Password

Results and Discussion

After the Password Analyzer was complete and ready to begin testing user's password strings, the software was put to action by selecting five users at random, from low level cyber-security skills up until highly trained specialists. I began to notice that just after the first attempt, the users began to take the suggestions provided by the analyzer and after running the software for the second time, 80% of the participant's password's strings became stronger. The code below displays how the score for the password string is obtained. There was only one participant that began strong with a score of 79% yet maintained a constant score of 79% through their first through third attempt. There was also an increase in 100% of the participants when comparing their first attempt to their fifth and final attempt. We can observe on the upcoming graph that all of the different user's password strings increased in strength after reading and applying the Password Analyzer's recommendations [3].



Results after Entering the User's Password Five Times while applying the recommendations

Conclusion

There has been an increase in cyber-security awareness throughout the years inside companies and organizations as well as general individuals [4]. With so many people acting and applying defenses against cyber-attacks, the number of victims all from large corporations to small industries will in fact be more prepared and less likely to become a vulnerable pray. Even with more and more entities absorbing knowledge and inquiring cyber-security tools, malicious hackers will not stop, yet they will continue to construct new and even more sophisticated methods of penetrating your systems. Therefore it is important to stay informed about any new and emerging threats and vulnerabilities.

Future Work

The two main areas where this project can be expanded for future work are for the following: general updates to the software, while the other area would be implementing this password analyzer to companies and organizations where employees will be forced to obtain a score of 80% or above for the systems to accept that users password string.

Acknowledgements

I Would like to thank Dr. Alfredo Cruz for contributing in the development of the Information Assurance and Security program
I would also like to thank my mentor Dr. Nelliud Torres for guiding me throughout every step of this project
And a special thanks to the graduate school staff for guiding me through this process

References

[1] Grassi, Paul, et al. "Digital Identity Guidelines."NIST Special Publication 800-63B, 1 Dec. 2017, pp. 1–69., doi:https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf.
[2] Samborski, John. "Hi-Tech Guessing Game: 350 Billion Passwords A Second". *Scientific Computing*, 2015, https://www.scientificcomputing.com/blog/2015/12/hi-tech-guessing-game-350-billion-passwords-second. Accessed 22 May 2018.
[3] "Create A Graph". *Nces.Ed.Gov*, 2018, https://nces.ed.gov/nceskids/createagraph/. Accessed 23 May 2018
[4] Bonderud, Douglas. "National Cyber Security Awareness Month: The 2017 Outlook". *Security Intelligence*, 2017, https://securityintelligence.com/national-cyber-security-awareness-month-the-2017-outlook/. Accessed 22 May 2018.